

COM00142M

Department of Computer Science

Advanced Programming

SUMMATIVE ASSESSMENT BRIEF

Author	Dr Seema Jehan
Assessment type	Summative assignment
Weighting	100%
Release	Week 3
Deadline	Monday following Week 8, 13:00 (UK time) *

* If this date falls on a UK public holiday or a University of York closure day, the submission date will change. Please check the submission point in the 'Assignments' area of the module in Canvas for the exact submission deadline.

Guidance for completing your assessment

At this level of study there may be multiple "right answers" to any given problem. You may have to make assumptions about ambiguous or incomplete information, interpret the information provided, or make justifiable decisions about approaches and techniques to apply. These should flow from the information provided, your understanding of the course materials and your further research around the subject area under assessment. Please state assumptions, interpretations and decisions, with justifications, clearly in your answer in order to assist the marker in understanding your work and giving you appropriate credit for it.

If you have any questions about this please direct them to your module tutor, in the first place, or cs-online-programmes@york.ac.uk

I. Module Learning Outcomes

The module learning outcomes (MLO's) for this module are as follows:

- MLO 1.** Demonstrate a critical understanding of the theory and application of advanced programming techniques.
- MLO 2.** Design and implement programs for real-world problems.
- MLO 3.** Communicate design decisions for the selection, storage and manipulation of data.
- MLO 4.** Critically evaluate the legal and ethical impact of software developments in real-world contexts.

This assessment addresses **all** the module learning outcomes listed above.

II. Assessment Background/Scenario

Your task is to design and develop a **prototype** application that demonstrates how data from the given data set can be formatted, reshaped and used to generate specific outputs. Your application can be a single programme or a collection of programmes that provide the equivalent functionality as described below.

Data Set (CSV)

The dataset contains online activity logs for 152 university students enrolled in a blended Computer Science course. The dataset is further split into three CSV files: "USER_LOG", "ACTIVITY_LOG", and "COMPONENT_CODES". The "USER_LOG" CSV consists of:

- Date
- Time
- User Full Name *Anonymized

The "ACTIVITY_LOG" CSV file contains:

- User Full Name *Anonymized
- Component
- Action

- Target

The “COMPONENT_CODES” CSV file contains:

- Component
- Code

Functional requirements

The application should provide the following basic functionality:

- A means to load the initial data set (CSV file(s) provided) and translate it into a suitable format – either XML, JSON or an entity relationship structure (not CSV).
- A means to back up the suitable format using either files or a database. This should preserve the current state of the data when the program is closed and make it available when the program is reopened.
- A process for cleaning and preparing the data set, managing inconsistencies, errors and missing values. Cleaning can be done at either the CSV stage or after you have translated the data set into a new format and is required to be done before you apply any of the data manipulations and outputs detailed below.
- A graphical user interface(s) for interacting with the data set(s) that enables the user to:
 - Load the initial data set (the CSV file(s)).
 - Apply the cleaning, transformation, **REMOVE** and **RESHAPE** to produce a prepared data set.
 - Load the prepared data set (from its translated format).
 - Manipulate the range of values used to generate **OUTPUT STATISTICS**, **GRAPHS** and perform **CORRELATION analysis**.
 - Use the prepared data set to generate **OUTPUT STATISTICS**, **GRAPHS** and **CORRELATION** results.

It should be assumed that this prototype application will be able to handle other sets of data generated from the same source, i.e. data with the same column and row structure in CSV format, but containing different values and anomalies. However, the application is not required to be generic (work with multiple unknown

data sets). Given this best practice regarding code reuse, encapsulation and a well-defined programming interface should be applied where applicable.

Data manipulation and outputs

Your prototype application needs to be able to perform the following actions on the data set, once it has been translated into your selected format. First, determine whether NumPy or Pandas is more appropriate for this dataset. Next, decide if it's more appropriate to split the data into manageable chunks before performing the following actions. Further, you should apply each of these actions in order, the later ones being the more challenging to achieve.

REMOVE: No outputs should include any data from Component: System, and Folder.

RENAME: The column "User Full Name *Anonymized" should be renamed as User_ID both in ACTIVITY_LOG and USER_LOG CSVs.

MERGE: Merge the suitable CSVs for analysing user interactions with each component.

RESHAPE: Reshape the data using pivot operation.

COUNT: The interactions for each user with the Component for each month. Add this new field to the new structure.

OUTPUT STATISTICS: Produce the mean, mode and median for the components: Quiz, Lecture, Assignment, Attendance, and Survey.

- a. For each month
- b. For the entire 13-week academic semester

OUTPUT CORRELATION: Produce a suitable graph that displays the following information from user interactions with the following components: Assignment, Quiz, Lecture, Book, Project, and Course. Determine if there is any significant correlation between the 'User_ID' and 'Component'. You will need to select an appropriate visualisation to demonstrate this.

Non-functional requirements

- The GUI interface must be able to provide appropriate feedback to confirm or deny a user's actions.
- The application must be able to handle internal and user-generated errors.

Technical requirements

- A. The application is built using Core Python from version 3.7 - 3.12.
- B. The application uses one or more of the advanced Application Programming Interfaces (API's) introduced on this module such as: NumPy, Pandas, Seaborn, Matplotlib. It should NOT use alternative API's for this functionality;

however, appropriate Python core libraries can be used to access/query a database.

- C. The application **MUST** run within the Anaconda environment using a Jupyter notebook.
- D. The application and its parts must not run concurrently, and must NOT use Python threads.

The requirements specified here are the constraints within which you need to produce your prototype application. They are not negotiable.

III. Assessment Task(s)

This assessment has two tasks:

- A. Design and implement a suitable prototype application that meets the specified requirements as either a single program, or a series of clearly identifiable programs. The program(s) submitted **MUST** be able to run under the constraints of the technical requirements section.
- B. Produce a report that addresses the questions below and demonstrates your approach to the design and development of your prototype application, clearly justifying the decisions you have made. You should support your discussion with appropriate reference to relevant sources using the correct citation and reference structure as indicated in the guide to [IEEE referencing system](#).

Where requested, you should select code samples from your software development that demonstrate specific algorithms and interactions. All code samples should be captured as images (screen shots), appropriately labelled, and presented in the appendix. You should refer to and discuss these within the context of each question. Do NOT include screenshots in the body of your report. For further guidance on using appendices, please see the 'Submission Formatting' page in Canvas.

Report contains 3 sections, as follows:

The report consists of three main sections, containing a series of questions to satisfy the learning outcomes. Each question has an indicative word count indicating what would be considered a reasonable response given the whole report. You may choose to redistribute this across questions; however, you must not exceed a total of **3,000 words** and a **maximum of 12 pages** in the appendices. There is no limit on the number of references you provide. For further guidance on word counts and the

required formatting of your report, please see the 'Submission Formatting' page in Canvas.

Section 1:

Theory supported by code samples (40%, 1,200 words plus up to 6 pages in the appendices)

Evidence for learning outcome: Demonstrate critical understanding of the theory and application of advanced programming techniques; design and implement programs for real-world problems. [MLO1, MLO2]

1a) [20 marks] Identify **ONE** part of your program design (such as processing the initial data set) that has the potential to be redesigned concurrently, using **Python Threads**. Clearly identify the program part and justify its selection and potential. Then discuss any specific issues that would need to be considered to refactor this part, and the wider impact of this refactoring on your whole program design. You should consider how data and/or communications will be passed between concurrent aspects, such as threads, and justify which Python constructs would support this redevelopment effectively.

It is expected that this question can be reasonably addressed within 600 words, with no more than 2 pages in the appendix for either pseudo code, diagrams or code samples that support your discussion. This section will require appropriate citations to achieve a pass.

1b) [20 marks] With specific reference to GUI interface constructs (such as text labels and buttons), and best practice regarding interface layouts, discuss how your GUI design and implementation supports **THREE** of the user interactions required by your prototype application. You should then justify your design decision for each, providing comparative examples to support your approach. You should aim to demonstrate as wide a range of interface constructs/layouts as your prototype application supports.

It is expected that this question can be reasonably addressed within 600 words, with no more than 4 pages in the appendix for GUI layout diagrams (wireframes OR screenshots) AND code samples that support your discussion. This section will require appropriate citations to achieve a pass.

Section 2:

Design decisions supported by code samples (40%, 1,200 words, up to 6 pages in the appendices)

Evidence for learning outcome: Communicate design decisions for the selection, storage and manipulation of data; design and implement programs for real-world problems.[MLO3, MLO2]

2a) [10 marks] With specific reference to the data manipulation requirements, **REMOVE** and **RESHAPE**, discuss your reasoning for your selected data format (JSON, XML, or entity relationship structure), and what advantages/disadvantages it has demonstrated in this context.

It is expected that this question can be reasonably addressed within 400 words with no more than 1 page of appendices for code samples, or data format samples. This section will require appropriate citations to achieve a pass.

NOTE: Failure to submit a functional program (or programs) in the Jupyter notebook format may result in a grade of zero for 2a only.

2b) [30 marks] For each of **OUTPUT STATISTICS, GRAPHS, and CORRELATION** discuss and demonstrate, via appropriate code samples and program output, the following:

- Any additional cleaning you have undertaken and justify it in the context of the relevant output(s). State clearly if you have carried out no additional cleaning, and justify why you chose not to do so.
- Explain why the APIs you selected for data analysis were chosen over other available options, focusing on how they are suited to producing the desired outputs.
- Provide a clear code example of how you have applied the selected API's to achieve each output.
- What you observe from each output and what conclusion/s you can draw from it, if any.

It is expected that this question can be reasonably addressed within 800 words, with no more than 5 pages in the appendix for code samples, and screenshots of output and visualisations that support your discussion.

Section 3:

Reflection on the ethical, moral and legal aspects (20%, 600 words)

Evidence for learning outcome: Critically evaluate the legal and ethical impact of software developments within real-world contexts. [MLO4]

3) [20 marks] Reflect on the ethical, moral and legal aspects of computing, as discussed in the module, and demonstrate an awareness of how these need to be considered in the role of a software engineer. Critically evaluate the following statement by building an effective 'for' or 'against' argument. This should be supported by the literature, using comparative examples, and recognition of the opposition's position where appropriate.

"The moderation of social media platforms by their owners/operators is robust, fair, and effective at removing problematic content. Consequently, software engineers should not be required to consider the ethical, moral or legal consequences of employing user-submitted social media content as training data for machine learning."

It is expected that this question can be reasonably addressed within 600 words. This section will require appropriate citations to achieve a pass.

IV. Deliverables

The appendices limit (12 pages) for this assessment supersedes that stated on the 'Submission Formatting' page in Canvas. Other than this, your assignment should be laid out following all other formatting guidelines that are specified in the 'Submission Formatting' page in Canvas.

You should submit two files as follows:

- A completed report answering the given questions as a single file in either .docx or .pdf format. This should **NOT** be included in the zipped file and should not exceed given word counts, or page limits.
- A single zipped file containing your program or programs. If a database has been used, you should produce a file dump of the data/table structure to include here. This should NOT contain the original data set.

Using a database:

Where you have opted to use an SQL or relational database (other than Mongo), include the following after your list of references:

- A. Name of database and link to download (install package)
- B. Version number of the database used
- C. The name of the Jupyter notebook that creates and populates the database
- D. The point in your code where local host and the port are set (make this clear)

You should make sure that your submitted code contains all the code required to set up and populate your database via a local host connection.

Referencing

You are required to use the [IEEE referencing style](#) for citing books, articles, and all other sources (such as websites) used in your assignment.

Good referencing is essential in order to meet the standards of academic integrity set by the University. All your sources must be acknowledged, regardless of whether you've included direct quotes or not. Visit your **Academic Integrity Tutorial module** in Canvas for additional guidance on effective referencing.

V. Marking Criteria

Section/Task	Criteria	Available marks
Section 1. Theory supported by code samples		
Functional program(s)	An implementation of your software design, using the specified platform(s), to demonstrate MLO2 as well as allow verification of your report discussion	
1a. Adaptation to a concurrent model	Appropriate concurrent mechanisms/constructs have been selected for the refactoring. These and potential issues/impacts have been discussed in the context of the given scenario and requirements.	20

1b. Implementing user interaction	Appropriate GUI constructs and layouts have been selected to support the required interactions. There is a clear rationale for their selection given best practice in GUI constructs and layout.	20
Section 2. Design decisions supported by code		
Functional program(s)	An implementation of your software design, using the specified platform(s), to demonstrate MLO2 as well as allow verification of your report discussion. (partially non-functioning code can still attract credit if it addresses the stated requirements and specification).	
a. Selected data format	<p>An effective format has been selected and a rational argument is presented for how it supports the nature of the data and the type of analysis required to produce the prototype applications requirements.</p> <p>Failure to submit a functional implementation may result in a grade of zero for this question only (2a)</p>	10
b. Generating outputs	Appropriate code constructs, internal data structures, visual representations have been selected and applied to achieve the given requirements. Considerations have been made for any anomalies within the data set. There is a clear justification for design decisions, and accurate observations made given the applications output.	30
Section 3. Reflection on ethics, morals and legal aspects		
Ethics, moral and legal	Clear and appropriate examples from the literature are used to build an effective argument to support a 'for' or 'against'	20

	position on the statement.	
	TOTAL	100

VI. Marking Criteria: Grade breakdown

Question 1a – Adaptation to a Concurrent Model [20%] [MLO1]

0-39%	Fail	<ul style="list-style-type: none"> • Either there is no clearly identified program part OR the part selected is not suitable to implement concurrently. • Either there is no suitable representation (pseudo code, diagram, or sample code) OR the representation is not referred to or discussed and provides little or no support for the intended refactoring. • There is little or no supporting evidence from the literature, OR the supporting literature is not relevant. • Appropriate constructs have been selected: There are no relevant constructs (given the selected program area and requirements of the brief), OR none of the constructs selected have been applied effectively. • Effective discussion of concurrent considerations: There are no relevant considerations (given the selected program area and requirements of the brief) OR the considerations identified are not discussed effectively.
40-49%	Marginal fail	<ul style="list-style-type: none"> • There is a clearly identified program which is not wholly suitable to implement concurrently. • There is a suitable representation (pseudo code, diagram, or sample code) which is not referred to or

		<p>discussed which could provide some support for the intended refactoring.</p> <ul style="list-style-type: none"> ● There is limited supporting evidence from the literature, little of which is relevant to the assessment brief. ● Appropriate constructs have been selected: There is at least ONE relevant construct (given the selected program area and requirements of the brief), but it has not been applied effectively. ● Effective discussion of concurrent considerations: There is at least ONE relevant consideration (given the selected program area and requirements of the brief), but it has only been discussed in a limited way.
50%-59%	Pass	<ul style="list-style-type: none"> ● There is a clearly identified program part which is suitable to implement concurrently. ● There is a suitable representation (pseudo code, diagram, or sample code) which is referred to, but may not be effectively discussed, and provides some support for the intended refactoring. ● There is a range of supporting evidence from the literature, which is somewhat relevant. ● Appropriate constructs have been selected: There is at least ONE appropriate construct (given the selected program area and requirements of the brief) which has been applied effectively to some degree. ● Effective discussion of concurrent considerations: There is at least ONE relevant consideration (given the selected program area and requirements of the brief) which has been discussed quite effectively.
60%-69%	Merit	<ul style="list-style-type: none"> ● There is a clearly identified program part which is suitable to implement concurrently.

		<ul style="list-style-type: none"> ● There is a suitable representation (pseudo code, diagram, or sample code) which is referred to in the discussion and provides support for the intended refactoring process. ● There is a range of supporting evidence from the literature, which is mostly relevant. ● Appropriate constructs have been selected: There are at least TWO appropriate constructs (given the selected program area and requirements of the brief) which have been applied in a mostly effective approach. ● Effective discussion of concurrent considerations: There are at least TWO relevant considerations (given the selected program area and requirements of the brief) which have been discussed in a mostly effective way.
70%-100%	Distinction	<ul style="list-style-type: none"> ● There is a clearly identified program part which is suitable to implement concurrently. ● There is a suitable representation (pseudo code, diagram, or sample code) which is referred to in the discussion and provides support for the intended refactoring process. ● There is a range of supporting evidence from the literature, which is relevant and effective. ● Appropriate constructs have been selected: There are at least TWO relevant considerations (given the selected program area and requirements of the brief) which have been applied effectively with consideration for wider requirements. ● Effective discussion of concurrent considerations: There are at least TWO relevant considerations

		(given the selected program area and requirements of the brief) which have been discussed clearly and effectively with consideration for wider impacts.
--	--	---

Question 1b – Implementing user interaction [20%] [MLO2]

0-39%	Fail	<ul style="list-style-type: none"> • Either there are no user interactions presented OR collectively they demonstrate little or no relevance, given the requirements of the brief. • Either there is no suitable representation (GUI layout diagrams, wireframes OR screenshots) OR the representation is not referred to or discussed and provides little or no support for the interaction design. • There is little or no supporting evidence from the literature, OR the supporting literature is not relevant or effective. • Appropriate constructs and layouts have been selected: Either none of the constructs and layouts are effective for any of the selected user interactions, OR collectively there are few effective constructs given the requirements of the brief and best practice. • Effective justification for the designs: Either no justification is provided for any of the selected user interactions OR collectively there is little evidence of justification, which is not rational and little or no support by reference to theory OR comparative examples.
40-49%	Compensable fail	<ul style="list-style-type: none"> • Either fewer than three user interactions have been presented OR none are wholly appropriate, but

		<p>collectively some parts are relevant given the requirements of the brief.</p> <ul style="list-style-type: none"> ● There is limited supporting evidence from the literature, little of which is relevant or effective. ● There is a suitable representation (pseudo code, diagram, or sample code) but it is not referred to or discussed, which could provide some support for the interaction design. ● Appropriate constructs and layouts have been selected: For at least ONE of the selected user interactions, OR collectively some parts of the selected constructs and layouts are effective in a limited way, given the requirements of the brief and best practice. ● Effective justification for the designs: For at least ONE of the selected user interactions, the justification for the approach is limited and not entirely rational and is supported in a limited way by either reference to theory OR comparative examples.
50%-59%	Pass	<ul style="list-style-type: none"> ● Three user interactions have been presented, of which at least ONE is appropriate given the requirements of the brief. ● There is a suitable representation (GUI layout diagrams, wireframes OR screenshots) which is referred to, but may not be effectively brought into the discussion, and provides some support for the interaction design. ● There is a range of supporting evidence from the literature, which is somewhat relevant and effective. ● Appropriate constructs and layouts have been selected: For at least ONE of the selected user interactions, constructs and layouts are somewhat effective given the requirements of the brief and

		<p>best practice.</p> <ul style="list-style-type: none"> ● Effective justification for the designs: For at least ONE of the selected user interactions, the justification for the approach is somewhat rational and is supported to some degree by appropriate references to theory and comparative examples.
60%-69%	Merit	<ul style="list-style-type: none"> ● Three user interactions have been presented, of which at least TWO are appropriate given the requirements of the brief. ● There is a suitable representation (GUI layout diagrams, wireframes OR screenshots) which is referred to and discussed, mostly supporting the interaction design. ● There is a range of supporting evidence from the literature, which is mostly relevant and effective. ● Appropriate constructs and layouts have been selected: For at least TWO of the selected user interactions, constructs and layouts are mostly effective, given the requirements of the brief and best practice. ● Effective justification for the designs: For at least TWO of the selected user interactions, the justification for the approach is mostly rational and is mostly supported by appropriate references to theory and comparative examples.
70%-100%	Distinction	<ul style="list-style-type: none"> ● Three user interactions have been presented, all are appropriate given the requirements of the brief. ● There is a suitable representation (GUI layout diagrams, wireframes OR screenshots) which is

		<p>referred to and discussed effectively, supporting the interaction design.</p> <ul style="list-style-type: none"> ● There is a range of supporting evidence from the literature, which is relevant and effective. ● Appropriate constructs and layouts have been selected: For all THREE of the selected user interactions, constructs and layouts are effective, given the requirements of the brief and best practice. ● Effective justification for the designs: For all THREE of the selected user interactions, the justification for the approach is rational and is clearly and effectively supported by appropriate references to theory and comparative examples.
--	--	---

2a – Selected data format [10%][MLO3]

0-39%	Fail	<ul style="list-style-type: none"> • Either no programs have been submitted OR samples have not been selected from the submitted program(s) OR the submitted program(s) is/are not functional. • There is little or no supporting evidence from the literature, OR the supporting literature is not relevant or effective. • Either the selected data format is not clearly stated OR the selected data format is not appropriate given the requirements of the brief. • Justification for selected data format: There is little or no justification for the data format given the requirements for the data manipulations, REMOVE and RESHAPE. This is supported poorly, if at all, by advantages or disadvantages over other available formats, which are unclear or have little relevance to the justification.
40-49%	Compensable fail	<ul style="list-style-type: none"> • There is a functional program from which samples have been selected. • There is limited supporting evidence from the literature, little of which is relevant or effective. • The selected data format is clearly stated but is not wholly appropriate given the requirements of the brief. • Justification for selected data format: There is a limited justification, which is not always clear for the data format given the requirements for the data manipulations, REMOVE and RESHAPE. This is supported in a limited way by advantages or disadvantages over other available formats, which are

		not clear or are tentative.
50%-59%	Pass	<ul style="list-style-type: none"> ● There is a functional program from which samples have been selected. ● There is a range of supporting evidence from the literature, which is somewhat relevant and effective. ● The selected data format is clearly stated and is appropriate given the requirements of the brief. ● Justification for selected data format: There is a somewhat rational justification for the data format given the requirements for the data manipulations, REMOVE and RESHAPE. This is supported by advantages and disadvantages, which are relevant to some degree, over other available formats.
60%-69%	Merit	<ul style="list-style-type: none"> ● There is a functional program from which samples have been selected. ● There is a range of supporting evidence from the literature, which is mostly relevant and effective. ● The selected data format is clearly stated and is appropriate given the requirements of the brief. ● Justification for selected data format: There is a mostly rational justification for the data format given the requirements for the data manipulation, REMOVE and RESHAPE. This is supported by mostly relevant advantages and disadvantages over other available formats.
70%-100%	Distinction	<ul style="list-style-type: none"> ● There is a functional program from which samples have been selected. ● There is a range of supporting evidence from the literature, which is relevant and effective.

		<ul style="list-style-type: none"> • The selected data format is clearly stated and is appropriate given the requirements of the brief. • Justification for selected data format: There is a clear and rational justification for the data format given the requirements for the data manipulation, REMOVE and RESHAPE. This is supported by relevant and effective advantages and disadvantages over other available formats.
--	--	---

2b – Generating outputs [30%][MLO2]

0-39%	Fail	<ul style="list-style-type: none"> • Additional cleaning: Either there is little or no indication that additional cleaning for any of the output requirements has been considered, OR the cleaning <i>has</i> been considered (if not explicitly stated), is inappropriate and there is little or no relevant justification for all the output requirements. • Selected API's: Either there are few or no effective API's OR there is little or no justification for the API's selected, for any of the output requirements. • Code Samples: Either there is little or no relevant code OR there is no reference to the code samples and there is little or no support provided for them in the discussion, for all the output requirements. • Observations and conclusions: Either there are few or no relevant screenshots, OR the screenshots do not reflect the observations and conclusions drawn OR the observations and conclusions drawn are inaccurate or inappropriate for all the output requirements.
40-49%	compensable Fail	<ul style="list-style-type: none"> • Additional cleaning: There is no clear statement as to whether cleaning has or has not been applied. The implied decision is tentatively appropriate for at least ONE of the output requirements, but is justified poorly.

		<ul style="list-style-type: none"> ● Selected API's: The API's for at least ONE of the output requirements are tentatively appropriate and are justified in a limited way in the context of the brief's requirements. ● Code Samples: The selected code samples are limited in their relevance for at least ONE of the output requirements. They are not referred to and only support the discussion in a limited way. ● Observations and conclusions: There are screenshots, which may be unclear, for at least ONE of the output requirements, which tentatively reflect the observations and conclusions drawn, which are not wholly accurate.
50%-59%	Pass	<ul style="list-style-type: none"> ● Additional cleaning: There is a clear statement as to whether cleaning has or has not been applied. This decision is somewhat appropriate for at least ONE of the output requirements, and is justified to some degree. ● Selected API's: The API's for at least ONE of the output requirements are somewhat appropriate and have been justified to some degree in the context of the brief's requirements. ● Code Samples: There are suitably selected code samples for at least ONE of the output requirements, which is referred to but may not support the discussion fully. ● Observations and conclusions: There are appropriate screenshots for at least ONE of the output requirements, which reflect the somewhat accurate observations and conclusions drawn.
60%-69%	Merit	<ul style="list-style-type: none"> ● Additional cleaning: There is a clear statement as to whether cleaning has OR has not been applied, which is mostly appropriate for at least TWO of the output requirements, and this is mostly justified in both cases. ● Selected API's: The API's for at least TWO of the output requirements are mostly appropriate and

		<p>have been mostly effectively justified in the context of the brief's requirements.</p> <ul style="list-style-type: none"> ● Code Samples: There are suitably selected code samples for at least TWO of the output requirements, which are referred to and mostly support the discussion. ● Observations and conclusions: There are appropriate screenshots for at least TWO of the output requirements, which reflects the mostly accurate observations and conclusions drawn.
70%-100%	Distinction	<ul style="list-style-type: none"> ● Additional cleaning: There is a clear statement as to whether cleaning has OR has not been applied, which is appropriate for all THREE of the output requirements, and effectively justified in all cases. ● Selected API's: The API's for at all THREE of the output requirements are appropriate and have been effectively justified in the context of the brief's requirements. ● Code Samples: There are suitably selected code samples for all THREE of the output requirements, which are referred to and clearly support the discussion. ● Observations and conclusions: There are appropriate screenshots for all THREE of the output requirements, which clearly reflects the accurate observations and conclusions drawn.

3 – Ethics, moral and legal [20%][MLO4]

0-39%	Fail	<ul style="list-style-type: none"> ● There is little or no supporting evidence from the literature, OR the supporting literature is not relevant or effective. ● There is no clear conclusion of 'for' or 'against' the statement.
-------	------	--

		<ul style="list-style-type: none"> ● Effective argument: There is little or no effective argument, with little clear consideration for either the moral, ethical, or legal aspects. There are few or no effective comparative examples, nor does the argument effectively present the opposition's position.
40-49%	Compensable Fail	<ul style="list-style-type: none"> ● There is limited supporting evidence from the literature, little of which is relevant or effective. ● There is an implied conclusion of 'for' or 'against' the statement. ● Effective argument: There is a limited argument that attempts to critically evaluate the statement drawing on at least ONE of the moral, ethical or legal aspects. This is not effectively supported by comparative examples, nor does it effectively present the opposition's position.
50%-59%	Pass	<ul style="list-style-type: none"> ● There is a range of supporting evidence from the literature, which is somewhat relevant and effective. ● There is a clear conclusion of 'for' or 'against' the statement. ● Effective argument: There is a somewhat relevant argument that critically evaluates the statement to some degree, drawing on at least ONE of the moral, ethical or legal aspects. This is somewhat effectively supported by comparative examples but may not effectively present the opposition's position.
60%-69%	Merit	<ul style="list-style-type: none"> ● There is a range of supporting evidence from the literature, which is mostly relevant and effective. ● There is a clear conclusion of 'for' or 'against' the statement. ● Effective argument: There is a mostly relevant argument that critically evaluates the statement in a reasonably effective way, drawing on at least TWO of the moral, ethical or legal aspects. This is mostly supported by comparative examples in an effective way, and there is some effective presentation of

		the opposition's position.
70%-100%	Distinction	<ul style="list-style-type: none"> ● There is a range of supporting evidence from the literature, which is relevant and effective. ● There is a clear conclusion of 'for' or 'against' the statement. ● Effective argument: There is a clear and relevant argument that critically evaluates the statement effectively, drawing on all THREE of the moral, ethical and legal aspects. This is effectively supported by comparative examples, and clearly presents the opposition's position.

VII. Assessment Submission

You will submit your assessment in the 'Assignments' area of the module in Canvas. Please check your Canvas module for the specific submission date for this assignment.

This assessment requires you to anonymously upload your submission to Canvas. If you are submitting multiple files, you must upload all files simultaneously to ensure that they are marked as a single submission. If you want to resubmit one component of your work, you need to re-upload all other files at the same time: every submission must include **all** files required by the assessment brief.

We recommend that you allow at least 30 minutes before the deadline to upload your submission, as failure to upload your assessment file within the allotted time is not admissible as an exceptional circumstance.

The webpage [How do I submit an online assignment?](#) provides further technical information on how to upload an assessment. The advice given here comes directly from Canvas. We do not recommend uploading assignments by mobile. We recommend you view the submission immediately after uploading your work to ensure the correct file has been submitted and no technical errors have occurred.

If you face any technical difficulties whilst trying to submit this assessment, then contact Canvas support on support@instructure.com or +44 80 0060 8442 (available 24 hours) in advance of the deadline. You should also email york-online-assessment@york.ac.uk as a matter of urgency to report the issue and receive further instruction.

VIII. Assessment Policies

This assessment is subject to the policies stated on the 'Summative Assessment Policies' page in Canvas. These policies include (but are not limited to):

- Academic Integrity and submission of student work to Turnitin
- Advice on anonymising your assessment
- Penalties for late submission
- Marking policy for multiple submissions
- The Fit to Sit / Submit policy
- Passing mark and module reassessment

Please ensure that you have read and understood these policies before starting the assessment.